المدرسة العليا للمواصلات بتونس
Ecole Supérieure des Communications de Tunis
Higher School of Communications of Tunis

Design Report
Soil Monitoring and Management System

Authors:
Fatma Ezzahra Ben Helal - Ines Mekki - Mariem Hadj Kacem - Rimel Hammami

Faculty Adviser:
Dr. Eng. Mohamed Becha Kaaniche

Contents

1 Introduction

1.1 Purpose of the Document

This document presents the **design phase** of the *Soil Monitoring and Management System*, focusing on its conceptual and structural representation through **UML diagrams**. The main objective is to translate the project specifications into clear visual models that describe how the system's components interact, behave, and are deployed. By providing these models, the design report ensures a shared understanding between the development, testing, and deployment teams, enabling a consistent and efficient implementation process.

1.2 Scope of the Document

This report focuses exclusively on the **UML modeling aspects** of the project. The document includes the following diagrams:

- **Class Diagram:** Define the system's structure, main entities, and their relationships.
- **Sequence Diagram:** Describes the flow of interactions between system components .
- **Activity Diagram:** Illustrates the overall workflow of the soil monitoring process.
- **State–Transition Diagram:** Depicts the life cycle and state changes of key system components, such as sensors.
- **Deployment Diagram:** Visualizes the physical architecture of the system, including hardware and software nodes.

1.3 System Overview

The *Soil Monitoring and Management System* is an **IoT-based precision agriculture solution** designed to help farmers make data-driven decisions. Its main components and functionalities include:

- **IoT Sensors:** Measure key soil and environmental parameters such as temperature, humidity, and NPK nutrient levels.
- **Raspberry Pi Gateway:** Collects sensor data and transmits it to the cloud using the MQTT communication protocol.
- **Middleware Layer:** Processes incoming data, manages storage, and facilitates communication with the machine learning models.
- **Machine Learning Models:** Analyze sensor data to provide intelligent recommendations for optimal crop selection and fertilizer usage.
- **Progressive Web Application (PWA):** Offers users real-time visualization of soil conditions, AI-generated insights, and alerts for abnormal values.

- **Geographical Contextualization (LBS):** Integrates location-based services to associate sensor data and ML predictions with precise geographical coordinates, supporting location-aware monitoring and decision-making.

By combining **IoT**, **cloud computing**, **machine learning**, and **LBS technologies**, the system aims to **improve resource efficiency**, **increase productivity**, and **promote sustainable agricultural practices**.

2 Class Diagram

This diagram defines the **static structure** of the system, including classes, their attributes, operations, and relationships.

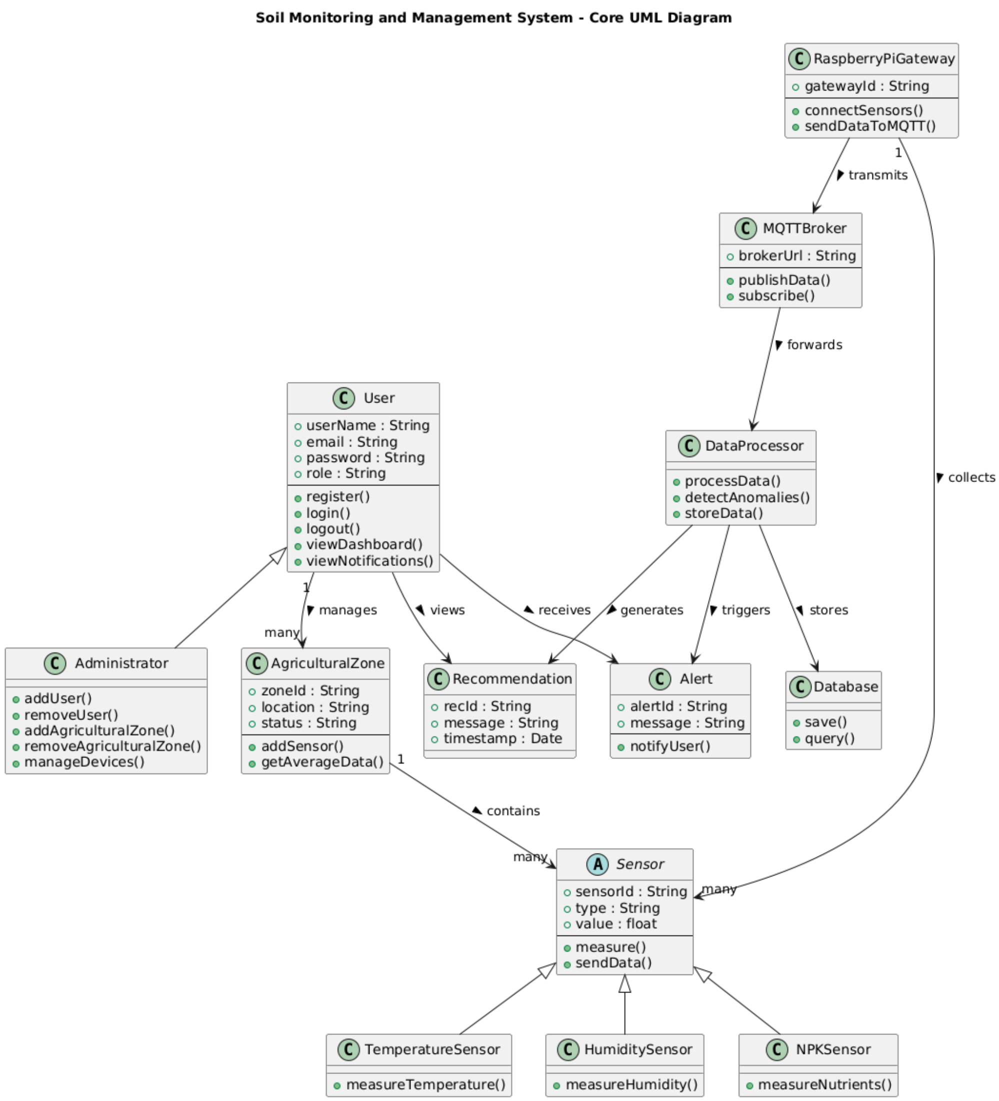

Figure 1: UML Class Diagram of the Soil Monitoring and Management System

3 Sequence Diagram

This section presents the dynamic interactions between system components across four main scenarios of the Soil Monitoring and Management System — from sensor data collection to displaying recommendations. Each diagram illustrates how objects communicate through message exchanges to achieve specific functionalities.

3.1 Data Flow from Sensors to Database Storage

This sequence shows the process of acquiring data from IoT sensors and storing it in the cloud database.

- Sensors (DHT22, NPK) measure temperature, humidity, and soil nutrient levels.
- Data is transmitted to the Raspberry Pi gateway.
- The gateway formats the data and publishes it via the MQTT protocol to the HiveMQ broker.
- The Middleware subscribes to the MQTT topic and receives the message.
- Middleware validates and preprocesses the data.
- The data is stored in the MongoDB database for further analysis.

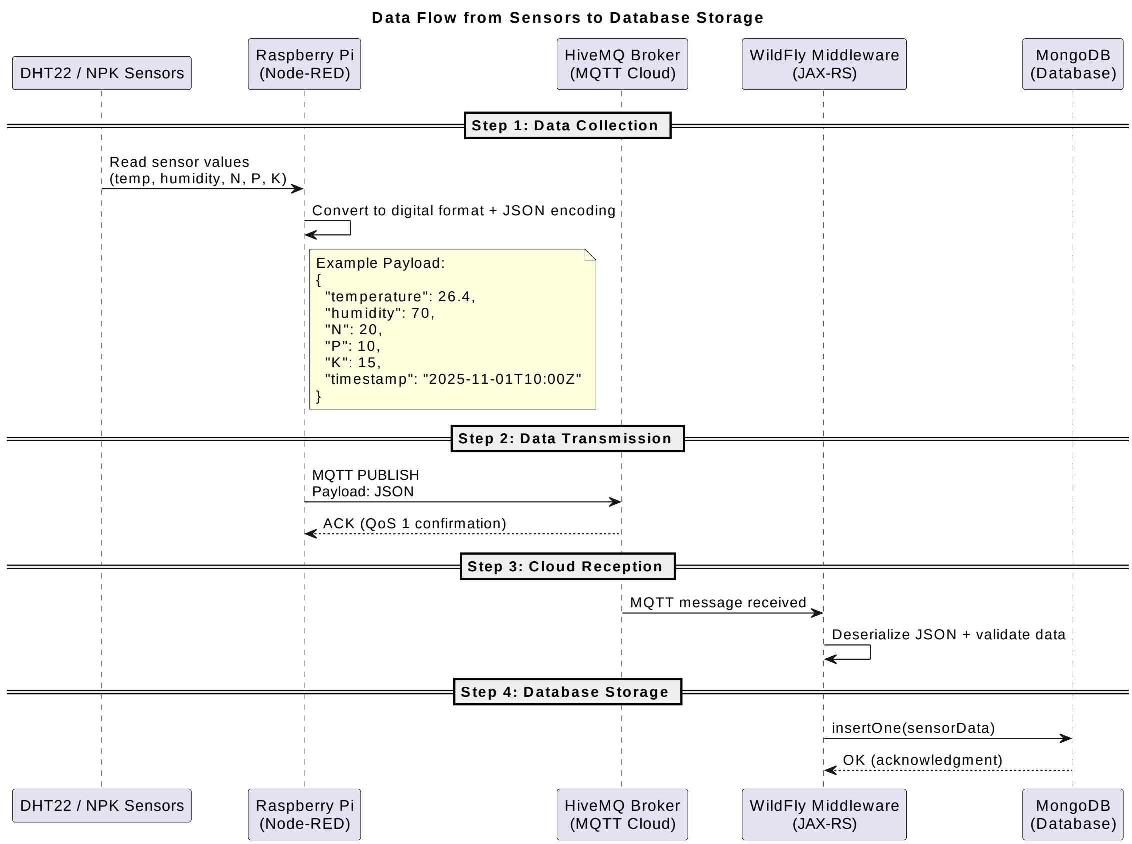

Figure 2: Data Flow from Sensors to Database Storage

3.2 Machine Learning Recommendation Generation

This diagram illustrates how the middleware interacts with the ML model to generate crop and fertilizer recommendations.

- Middleware retrieves recent soil and environmental data from MongoDB.
- Middleware sends this data to the ML Model service.
- The ML Model processes the input using trained algorithms (CatBoost for crops, Random Forest for fertilizers).
- The ML Model returns recommendations (best crop, fertilizer type, and quantity).
- Middleware stores these recommendations and sends them to the PWA for user access.

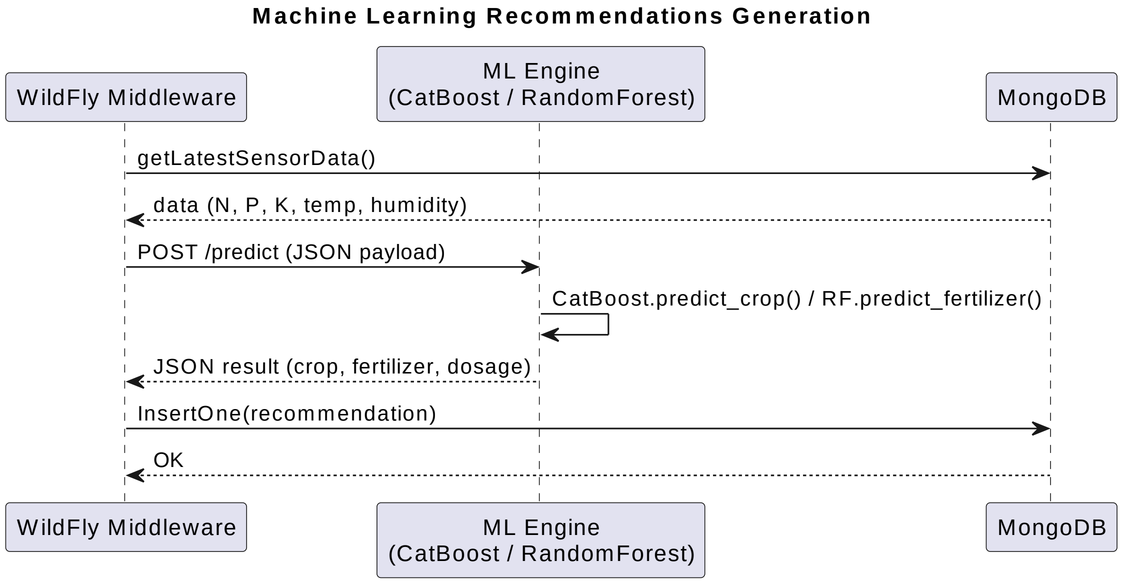

Figure 3: Machine Learning Recommendation Generation

3.3 Real-Time Alert Detection and Notification

This sequence explains how the system detects abnormal readings and notifies the user instantly.

- Sensor readings are received by the Middleware.
- Middleware compares values against predefined thresholds (e.g., temperature too high, low NPK levels).
- If an anomaly is detected, Middleware generates an alert event.
- The alert message is sent to the user through the WebSocket connection.
- The PWA displays a visual and/or audible alert notification on the user dashboard.

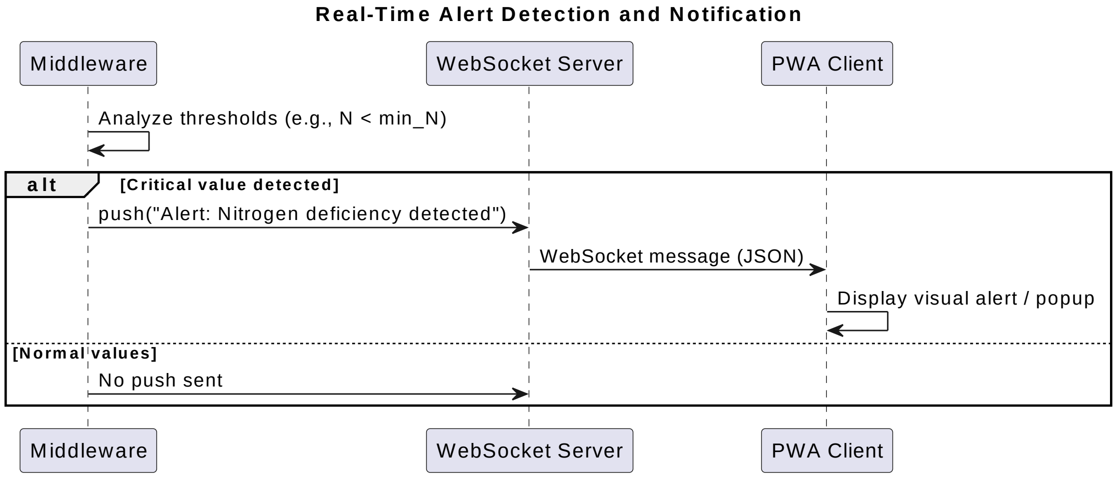

Figure 4: Real-Time Alert Detection and Notification

3.4 Data Visualization on the PWA

This sequence describes how the PWA retrieves and displays real-time and historical data to the user.

- User requests to view soil data through the PWA interface.
- The PWA sends a data request to the Middleware via REST API.
- Middleware queries MongoDB to retrieve the requested records.
- Middleware sends the data response to the PWA.
- The PWA visualizes the data through dynamic charts and tables.

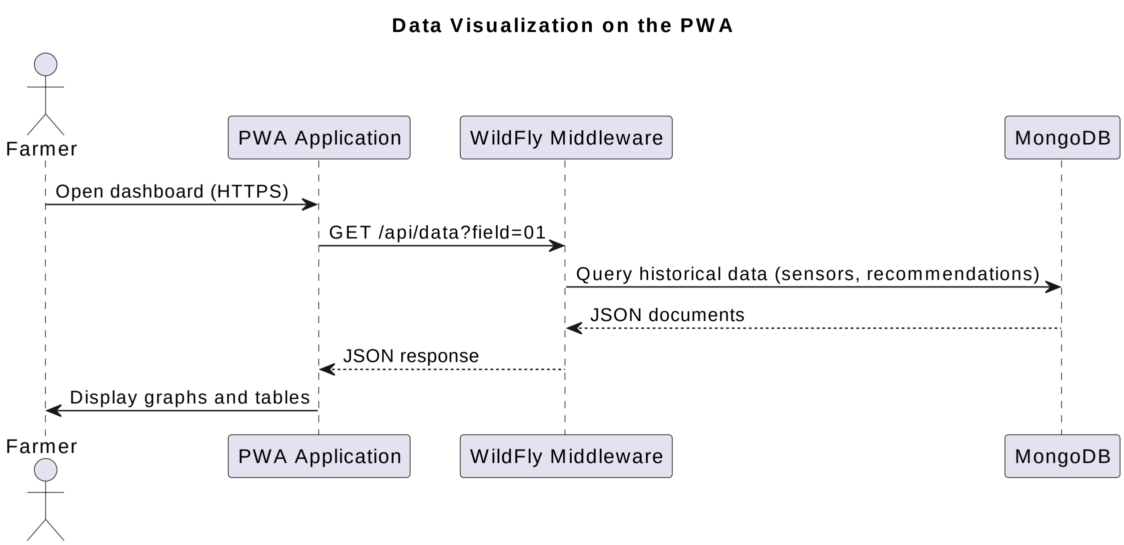

Figure 5: Data Visualization on the PWA

3.5 Summary

These four sequence diagrams collectively represent the core operations of the system — from raw data collection to intelligent analytics, alerts, and user interaction — demonstrating the seamless integration of IoT, machine learning, and real-time web technologies.

4 Activity Diagrams

This section presents the activity diagrams that describe the key operational workflows of the **Soil Monitoring and Management System**. Two main processes are modeled: (1) Data Acquisition and Transmission, and (2) Machine Learning Recommendation Generation.

4.1 Data Acquisition and Transmission

The first diagram illustrates how soil and environmental data are collected from sensors and transmitted to the cloud database. It emphasizes the interactions between the sensors, the Raspberry Pi gateway, the MQTT broker, and the middleware responsible for data validation and storage.

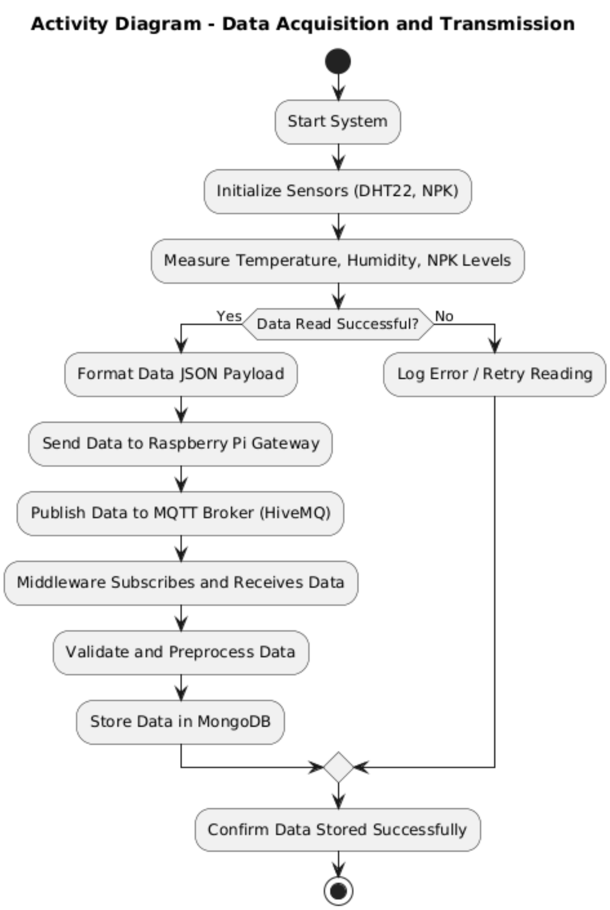

Figure 6: Activity Diagram – Data Acquisition and Transmission

4.2 Machine Learning Recommendation Generation

The second diagram represents the process of generating AI-based recommendations for crops and fertilizers using machine learning models. It highlights how the middleware communicates with the ML model and the PWA to deliver intelligent insights to users.

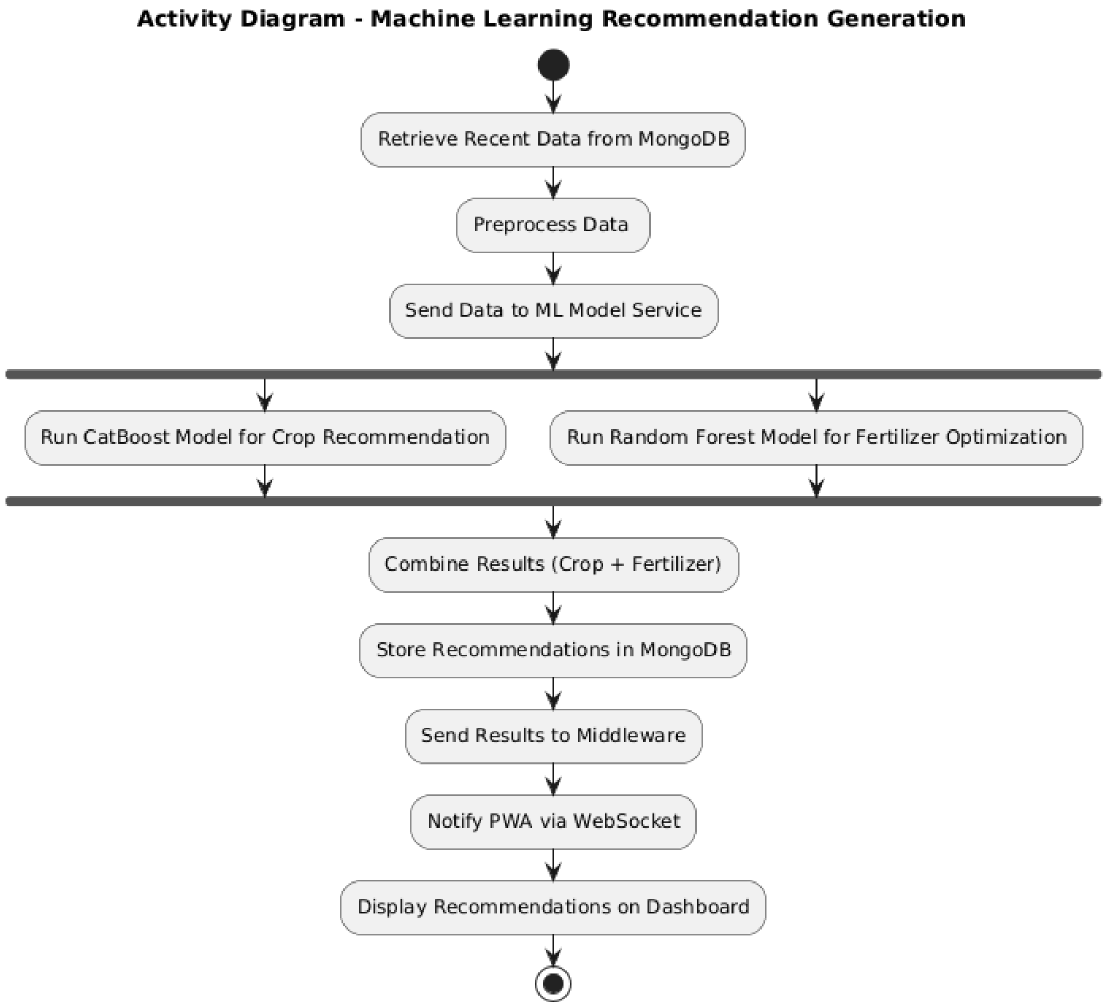

Figure 7: Activity Diagram – Machine Learning Recommendation Generation

These two activity diagrams together describe the operational backbone of the system. The first focuses on data collection and transmission, while the second emphasizes analytical intelligence and recommendation generation, illustrating the complete data-driven loop of the solution.

5 State–Transition Diagram

IoT Sensor State Machine

The IoT Sensor (DHT22 or NPK) follows a cyclical process of data acquisition, transmission, and error handling. The state–transition diagram in Figure **??** illustrates its life cycle.

Explanation of States:

- **Idle:** The sensor is powered on but inactive. It waits for initialization or calibration to start. This is the resting state before measurement begins.
- **Initializing:** The sensor performs internal checks and calibration routines. Successful calibration transitions it to the measurement phase.
- **Measuring:** The sensor collects temperature, humidity, or NPK data from the soil. The data is then validated and formatted for transmission.
- **Transmitting:** The validated data is sent via the MQTT protocol to the Raspberry Pi gateway. A successful transmission returns the system to *Idle* for the next reading.
- **Error:** Triggered when a failure occurs during calibration, reading, or transmission. The sensor attempts recovery or waits for a manual reset.

Transition Logic:

- `PowerOn / startCalibration():` – Activates and starts calibration.
- `calibrationDone / beginMeasurement():` – Moves to data collection.
- `dataReady / sendMQTT():` – Publishes data to the cloud.
- `transmissionSuccess / standby():` – Loops back to idle.
- `error / reinitialize():` – Attempts to recover from fault.

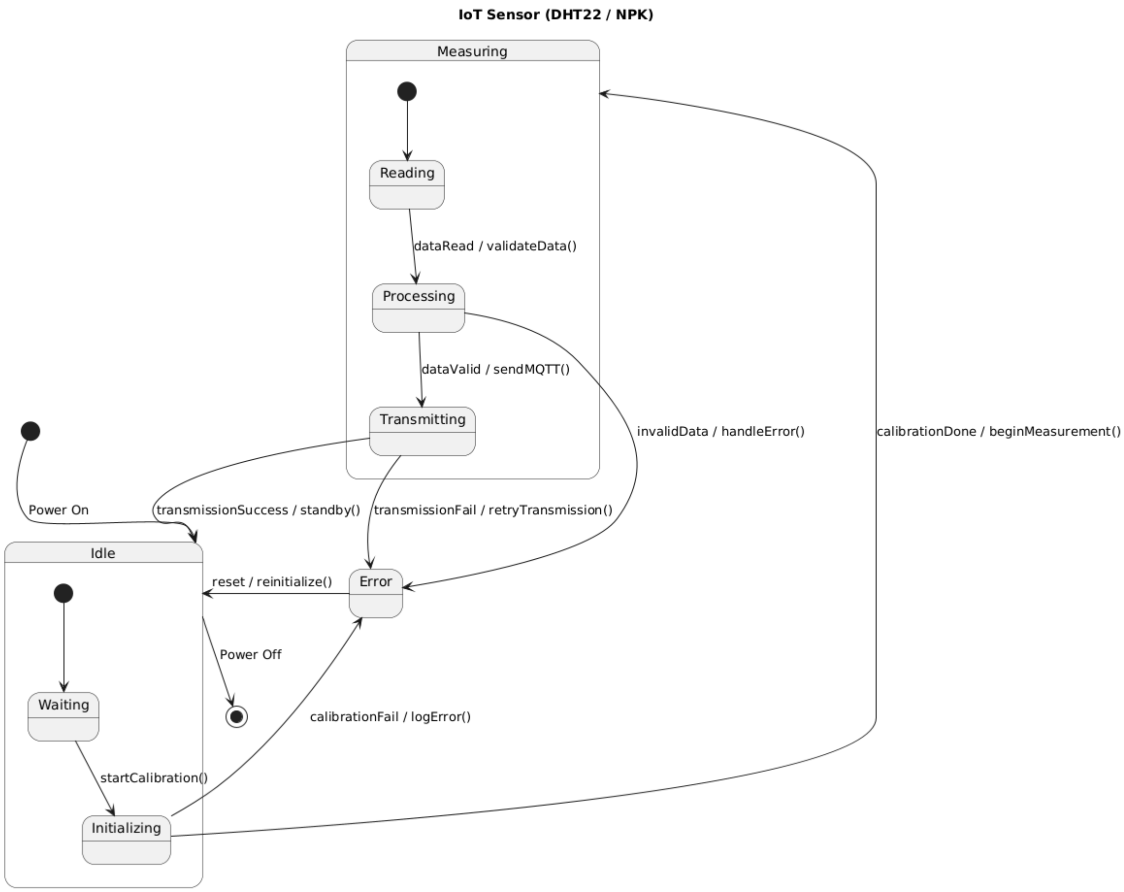

Figure 8: UML State–Transition Diagram of the sensor component

This state machine ensures a reliable and repeatable operational cycle for each IoT sensor.

6 Deployment Diagram

The deployment diagram visualizes the **physical architecture** and how software components are distributed across hardware nodes. **Nodes:**

- **Mobile Device:** Laptop or smartphone running PWA.
- **Cloud Server:** WildFly + MongoDB + ML services.
- **Edge Device:** Raspberry Pi 4 with Node-RED.
- **Sensors:** DHT22, NPK sensor modules.

Connections:

- HTTPS/REST API (PWA ↔ Cloud API)
- MQTT (Raspberry ↔ Cloud)

- GPIO /Serial (Sensors ↔ Raspberry)

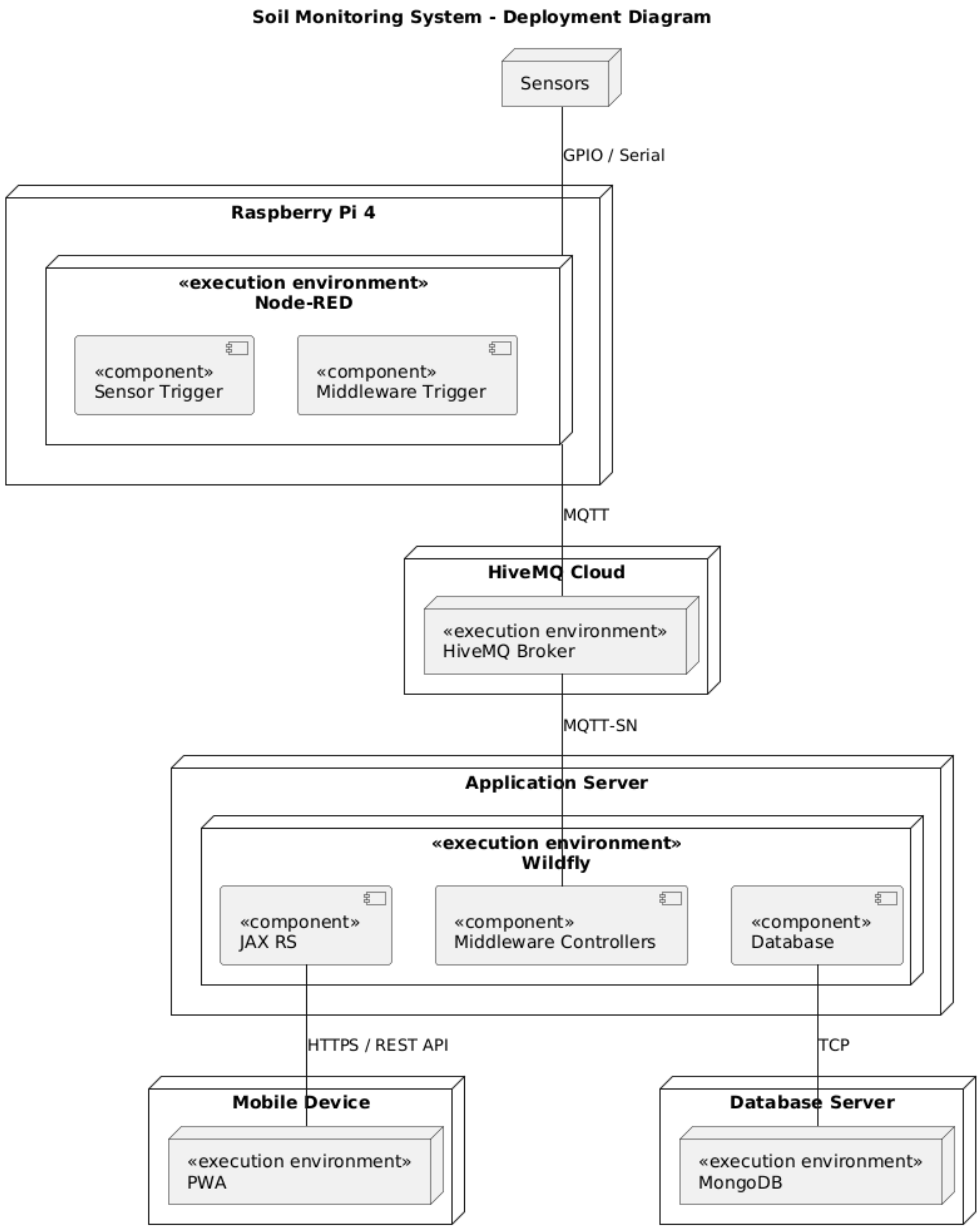

Figure 9: UML Deployment Diagram of the system architecture

7 Conclusion

The Soil Monitoring and Management System provides a unified, UML-based design integrating IoT, cloud computing, machine learning, and location-based technologies for precision agriculture. The presented diagrams define the system's architecture, ensuring efficient data flow, intelligent analysis, and real-time decision support. This design lays a solid foundation for implementation, promoting productivity, sustainability, and location-aware agricultural management.